EECS 455: Solutions to Problem Set 6

1. Consider a communication system that transmits one of 16 signals in 7 dimensions. The signals are

$$\begin{aligned}
s_0 &= \sqrt{E}(+1,+1,+1,+1,+1,+1,+1)\\
s_1 &= \sqrt{E}(+1,+1,+1,-1,+1,-1,-1)\\
s_2 &= \sqrt{E}(+1,+1,-1,+1,-1,-1,+1)\\
s_3 &= \sqrt{E}(+1,+1,-1,-1,-1,+1,-1)\\
s_4 &= \sqrt{E}(+1,-1,+1,+1,-1,-1,-1)\\
s_5 &= \sqrt{E}(+1,-1,+1,-1,-1,+1,+1)\\
s_6 &= \sqrt{E}(+1,-1,-1,+1,+1,+1,-1)\\
s_7 &= \sqrt{E}(+1,-1,-1,-1,+1,-1,+1)\\
s_8 &= \sqrt{E}(-1,+1,+1,+1,-1,+1,-1)\\
s_9 &= \sqrt{E}(-1,+1,+1,-1,-1,-1,+1)\\
s_{10} &= \sqrt{E}(-1,+1,-1,+1,+1,-1,-1)\\
s_{11} &= \sqrt{E}(-1,+1,-1,-1,+1,+1,+1)\\
s_{12} &= \sqrt{E}(-1,-1,+1,+1,+1,-1,+1)\\
s_{13} &= \sqrt{E}(-1,-1,+1,-1,+1,+1,-1)\\
s_{14} &= \sqrt{E}(-1,-1,-1,+1,-1,+1,+1)\\
s_{15} &= \sqrt{E}(-1,-1,-1,-1,-1,-1,-1)
\end{aligned}$$

Four bits are transmitted each with energy $7E$ so the energy per bit is

$$E_b = 7E/4.$$

We will consider two receivers.

(a) The first receiver is the optimal receiver for minimizing the probability of signal error (choosing the wrong signal). For this receiver we want to determine the probability of bit error. Assume the 4 bits that are transmitted are represented in binary and the binary number determines the signal. That is the bits 0000 represent 0 and are mapped to s_0. The bits 0001 represent 1 and are mapped to s_1. The bits 0101 represent 5 and are mapped to s_5. Simulate this communication system in additive white Gaussian noise (power spectral density $N_0/2$ to determine the bit error probability. Plot P_e versu $E_b/N_0(dB)$ as $E_b/N_0(dB)$ varies from 0 dB to 7dB.

Solution: The plot and code are shown below.

```
clear all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                          %
%                 Simulation Parameters                    %
```

```
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


ncount=input('Number of errors = ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
%                 Setup the Simulation                                  %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for m=1:25
EbN0dB(m)=-4+(m-1)/2
EbN0(m)=10^(EbN0dB(m)/10);
Eb=1;
E=Eb*4/7;                  % Relation between energy per code symbol and
N0=Eb/EbN0(m);             % Noise power
sigma=sqrt(N0/2);

s0  =  sqrt(E)*[+1, +1, +1, +1, +1, +1, +1];
s1  =  sqrt(E)*[+1, +1, +1, -1, +1, -1, -1];
s2  =  sqrt(E)*[+1, +1, -1, +1, -1, -1, +1];
s3  =  sqrt(E)*[+1, +1, -1, -1, -1, +1, -1];
s4  =  sqrt(E)*[+1, -1, +1, +1, -1, -1, -1];
s5  =  sqrt(E)*[+1, -1, +1, -1, -1, +1, +1];
s6  =  sqrt(E)*[+1, -1, -1, +1, +1, +1, -1];
s7  =  sqrt(E)*[+1, -1, -1, -1, +1, -1, +1];
s8  =  sqrt(E)*[-1, +1, +1, +1, -1, +1, -1];
s9  =  sqrt(E)*[-1, +1, +1, -1, -1, -1, +1];
s10  =  sqrt(E)*[-1, +1, -1, +1, +1, -1, -1];
s11  =  sqrt(E)*[-1, +1, -1, -1, +1, +1, +1];
s12  =  sqrt(E)*[-1, -1, +1, +1, +1, -1, +1];
s13  =  sqrt(E)*[-1, -1, +1, -1, +1, +1, -1];
s14  =  sqrt(E)*[-1, -1, -1, +1, -1, +1, +1];
s15  =  sqrt(E)*[-1, -1, -1, -1, -1, -1, -1];

allsignals=[s0; s1; s2; s3; s4; s5; s6; s7; s8; s9; s10; s11; s12; s13;
nerrors=0;
nbsim=0;
while(nerrors < ncount)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
```

```
%               Generate data and signals                 %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

b=round(rand(1,4));
index=(b(4)+2*b(3)+4*b(2)+8*b(1))+1;
s=allsignals(index,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                         %
%                   Add Noise                             %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

noise=sigma*randn(1,7);
      rcvd=s+noise;

%FOR HARD DECISIONS DECODING ADD THIS LINE
        rcvd=sign(rcvd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                         %
%           Decode the received signal                %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        [corr,l]=max(rcvd*allsignals');
lhat=l-1;
        bhat(4)=mod(lhat,2);
lhat=floor(lhat/2);
bhat(3)=mod(lhat,2);
lhat=floor(lhat/2);
bhat(2)=mod(lhat,2);
lhat=floor(lhat/2);
bhat(1)=mod(lhat,2);
        nerrors=nerrors+sum(abs(sign(bhat-b)));
        nbsim=nbsim+4;
if (mod(nbsim,5000)==0)
peb(m)=nerrors/nbsim;
format('long')
[EbN0dB(m),nerrors,nbsim,peb(m)]
                save s095data
end;
   end

peb(m)=nerrors/nbsim;
```

```
semilogy(EbN0dB,peb)
axis([-4 12 0.00000001 1])
grid on
hold on
xlabel('E_b/N_0 (dB)')
ylabel('P_{e,b}')
save s095data
end
```

(b) The second receiver computes the decision variable corresponding to each dimension $\tilde{r}_i = \text{sign}(r_i)$ and then finds the signal vector that has the fewest differences (number of positions where an error occurs). Simulate this system as well. Determine the bit error probability as a function of the energy per information bit. Plot P_e versu $E_b/N_0(dB)$ as $E_b/N_0(dB)$ varies from 0 dB to 8dB.